

基于 XML 的数据交换技术在电子商务中的应用

Application of Data Exchange Technology Based on XML in eBusiness

刘琴 (上海华东政法学院信息技术中心 200042)

王霞 (上海金融学院信息管理系 201209)

摘要: XML 近来已成为科学与业务应用中数据表示的标准以及 Web 上应用间数据交换的标准。在需求与技术的驱动下,各数据库厂商纷纷进行 XML 功能扩充。本文介绍了利用商业 DBMS 提供的 XML 扩充功能开发的一个在电子商务 B2B 中的应用:网上书店零售商系统以及供应商系统。

关键词: 数据交换 XML 关系数据库 电子商务

1 商业 DBMS 中的 XML 技术

本文主要介绍 DB2 和 SQL Server 中 XML 的技术。

IBM DB2 XML Extender^[2] 部件 (已被打包在 DB2 UDB7.1 中) 用来实现整体或拆分存储 XML 文档、从现存的关系数据生成 XML 文档。它有两种可选的工作方式: (1) XML Column: 以用户定义类型 (XMLCLOB、XMLVARCHAR、XMLFILE) 中的任一种来存储整个 XML 文档到关系表的某列中; (2) XML Collections: 允许用户拆分 XML 文档到 DB2 的表中或生成 XML。映射过程中用一个 XML 格式的 Data Access Definition (DAD) 文档来指定映射规则。

现 XML 的存储和出版,为熟悉 XML 的用户提供了基于 XML 的方法——在关系数据上建立 XML 视图。

2 系统的总体设计

如图 1 所示,本实例包括两个站点:网上书店零售站点以及供应商站点。

零售商主要负责从客户那里获取定单,为了避免资金积压,零售商处不存货,故而它把所有付款以后的客户定单以相互约定的统一的 XML 的形式 (即遵守同一个 DTD, 定单的 DTD 如图 2 所示) 发送给供应商,供应商读取有用的数据并与供应商数据库中的数据进行比较后,再将订购确认信息返回给供应商。除了定单的发送外,供应商也要把商品 (主要包括图书和 CD) 的信息及时地以 XML 的形式 (商品信息的 DTD 如图 3 所示) 发送给零售商。

3 系统实现中的关键技术

本实例由两台互连的 PC 机组成,分别作为零售商和供应商的 Web 服务器。

零售商: 操作系统为 Windows2000; Web 服务器软件为 Tomcat4.0; 数据库为 DB2 UDB7.1, 存储和出版 XML 采用 DB2 XML Extender, 它可从 <ftp://ftp.software.ibm.com/ps/products/db2xml/fixes/fixpaks/v7fp7/db2xmlintfp7.zip> 下载, 下载后先安装, 然后将数据库 (bookshop) 绑定到 XML Extender, 并通过 DB2 的 GUI 界面 enabling 此数据库;

供应商: 操作系统为 Windows2000 Server; Web 服务器软件为 WebLogic7.0; 数据库为 SQL Server2000, SQL Server 扩展了 SQL 语言来支持存储和出版 XML, 没有专门的工具;

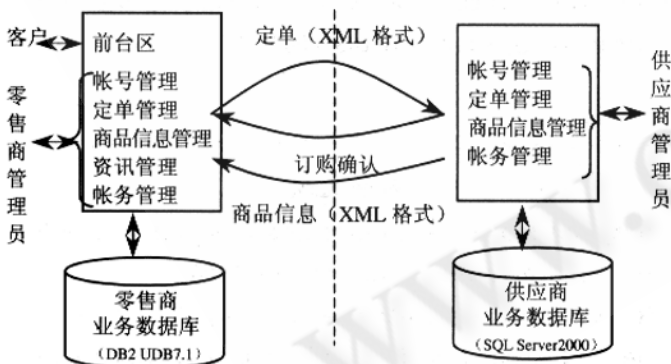


图 1 系统体系结构

IBM 新发行的 DB2 V8^[3] 中引入了 SQL/XML 标准 (制定在 SQL 和 XML 之间如何进行映射) 的一些函数, 主要有三类: (1) 创建元素与属性的函数, 如 XMLELEMENT、XMLATTRIBUTES; (2) 聚集函数, 如 XMLAGG; (3) 指定复杂嵌套结构的子查询。它们能更好地支持 XML 的出版。

SQL Server2000 为熟悉关系世界的用户提供了扩展了的 SQL 语句 (FOR XML 子句和行集函数 OpenXML) 来实

```

<! ELEMENT order ( customer, itemlist, total
price, fulfillment, statement) > <! ATTLIST order
order_id CDATA #REQUIRED >
<! ATTLIST order order_date CDATA #REQUIRED
>
<! ELEMENT customer ( name, add, zip, email?,
phone1, phone2?) >
<! ELEMENT name( #PCDATA) >
<! ELEMENT add( #PCDATA) >
<! ELEMENT zip( #PCDATA) >
<! ELEMENT email( #PCDATA) >
<! ELEMENT phone1( #PCDATA) >
<! ELEMENT phone2( #PCDATA) >
<! ELEMENT itemlist( item+ ) >
<! ELEMENT item ( title, author, publisher, pub
date?, quan) >
<! ELEMENT title( #PCDATA) >
<! ELEMENT author( #PCDATA) >
<! ELEMENT publisher( #PCDATA) >
<! ELEMENT pub_date( #PCDATA) >
<! ELEMENT quan( #PCDATA) >
<! ELEMENT total_price( #PCDATA) >
<! ELEMENT fulfillment ( pay_mode, pay_date,
send_mode, send_date?) >
<! ELEMENT pay_mode #PCDATA >
<! ELEMENT pay_date #PCDATA >
<! ELEMENT send_mode #PCDATA >
<! ELEMENT send_date #PCDATA >
<! ELEMENT statement #PCDATA >

```

图 2 order.dtd

```

<! ELEMENT product ( title, author, description,
price, publisher, pub_date?) >
<! ELEMENT title( #PCDATA) >
<! ELEMENT author( #PCDATA) >
<! ELEMENT description( #PCDATA) >
<! ELEMENT price( #PCDATA) >
<! ELEMENT publisher( #PCDATA) >
<! ELEMENT pub_date( #PCDATA) >

```

图 3 product.dtd

开发语言及工具: 用 XML、JSP、Servlet 等语言实现,

JSP、Servlet 的开发工具为 Jbuilder7。

3.1 零售商系统

本文将介绍主要的核心功能: 审核定单、处理商品信息。

3.1.1 审核定单

当收到用户付款后, 首先由零售商管理员核对定单、填写支付日期, 然后单击“发送”, 系统将调用 RetailerOrder.java 程序, 该程序主要分为如下几个部分:

(1) 支付日期的保存: 将付款日期保存到订单表 orders 中;

(2) XML 生成: 利用 DB2 XML Extender (XML Collections 方式下), 编写 SQL_stmt 风格的 DAD, 然后调用存储过程 db2xml.dxxGenXML() 依据表 orders, orders_detail, products 生成 XML, 不过此时的 XML 还是以 XMLVARCHAR 的类型存放在结果表 result 列 orderxml 中, 还需用如下的 SQL 语句从结果中抽取所需 XML。

```
rs = st.executeQuery
```

```
("select db2xml.extractVarchar(result.orderxml, '/
orderlist') from result" +
```

```
"where db2xml.extractVarchar(result.orderxml, '/
orderlist/order/@order_id') = " + id);
```

```
orderXML = rs.getString(1);
```

```
//字符串变量 orderXML 保存定单
```

(3) 定单发送: 将字符串变量 orderXML 中的 XML 数据作为参数以 POST 请求的方式发往供应商方 Servlet——SupplierOrder.java 处理。本实例中使用了 Jason Hunter 实现的 com.oreilly.servlet.HttpMessage 类来完成这个任务, 代码如下:

```
URL url = new URL(SUPPLIER_URL);
```

```
//SUPPLIER_URL 保存供应商的 URL
```

```
HttpMessage msg = new HttpMessage(url);
```

```
Properties p = new Properties();
```

```
p.put("order", orderXML);
```

```
//将 XML 数据赋予 order 属性
```

```
InputStream in = msg.sendPostMessage(p);
```

```
//发送属性表 p 中的数据并接收响应数据, 如果发送不成功, 则抛出 IOException,
```

```
//由方法 sendPostMessage 捕获后, 通知管理员重新发送。
```

(4) 处理响应: 响应信息是字符串, 如果交易失败, 供应商返回失败原因, 则将失败原因添加到表 orders 的 statement 列中, 并且将失败原因通知管理员; 如果交易成功, 供应商返回送货日期, 则把送货日期保存到表 orders 的 send_date 列中, 并且通知管理员。

3.1.2 处理商品信息

处理来自供应商的商品信息, 实现这一功能的是 Re-

tailerProduct.java, 商品数据以 XML 的形式通过 POST 请求作为 product 属性的值到达零售商 Servlet, 同时还有 insOrDel 属性, 此属性告诉零售商是要删除还是增加商品信息。这些属性可通过 request.getParameter() 方法读取。

如果是增加信息, 则在 DB2 XML Extender 中的 XML Collections 方式下, 编写 RDB_node 风格的 DAD (与 SQL _stmt 风格的 DAD 相似, 只是需要指出存储 XML 的表的主键和列类型), 然后调用存储过程 db2xml.dxxinsertXML() 把 XML 形式的商品数据保存到表 products 中; 如果要删除信息, 则在 XML Column 方式下先把该 XML 存放到临时表 productTem 列 productxml 中, 然后利用系统提供的 UDF 函数处理。具体 SQL 语句如下:

```
delete
from products
where
title = db2xml.ExtractVarchar ( productTem. productxml, '/product /title)
and author = db2xml.extractVarchar
(productTem.productxml, '/product /author);
```

3.2 供应商系统

本文将介绍主要的核心功能: 处理定单、增/删商品信息。

3.2.1 处理定单

SupplierOrder.java 实现供应商的定单处理。定单数据通过 POST 请求作为 order 属性的值到达供应商, 通过 request.getParameter("order") 便可读取其数据, 将该数据保存在字符串 order 中, 利用如下的程序段, 判断所购商品的数量能否被满足, 如果满足, 交易成功, 那么把该定单数据添加到数据库中、修改相应商品的库存量, 并使用响应对象 HttpServletResponse 将发货日期返回; 如果交易失败, 同样用对象 HttpServletResponse 将失败原因返回。

3.2.2 增/删商品信息

当供应商要增加商品信息时, 先把增加的信息保存到 SQL Server 数据库, 然后再发送该商品信息到零售商; 当供应商要删除商品信息时, 先从数据库中将商品从保存商品信息的表 product_supplier 中删除, 再把该商品信息发送给零售商。

实现发送功能的 Servlet 是 SupplierProduct.java 程序, 此程序与 RetailerOrder.java 类似。

首先是 XML 的生成。

接着是商品信息和增/删信息的发送, 代码与 RetailerOrder.java 类似, 只是除了 product 属性外, 还要增加一

个 insOrDel 属性, 用于表明 product 属性是要增加的商品信息还是要删除的信息。

```
URL url = new URL(RETAILER_URL);
//RETAILER_URL 保存零售商的 URL
HttpMessage msg = new HttpMessage(url);
Properties p = new Properties();
p.put("product", productXML);
//将 XML 数据赋予 product 属性
p.put("insOrDel", insOrDel);
//将增/删信息赋予 insOrDel 属性
InputStream in = msg.sendPostMessage(p);
//发送属性表 p 中的数据并接收响应数据
out.println("商品信息已发送"); //通知管理员
```

4 结束语

本文给出了数据交换技术在 B2B 中的具体应用实例, 从开发角度看, 在 XML 扩展方面 DB2 比 SQL Server 要更完善些, 例如: DB2 把生成的 XML 以 XMLVARCHAR 类型存放到临时表中, 应用程序可以调用各种用户定义函数 (UDF) 方便地提取出各种数据类型的数据或者把 XML 以文件的方式存放, 而 SQL Server 把生成的 XML 以字符串的形式存放在临时表中, 如果想要获取 XML 文件, 还需要另外编程来实现; 不过, SQL Server 由于直接扩展了 SQL, 所以实现起来要比 XML Extender 容易。

目前在 XML 数据和关系数据之间还没有标准的映射方式, 不过这一现象会随着由 SQLX 组织着手制定的 SQL/XML 标准的出台而改变。

参考文献

- 1 Ceri S, Fraternali P, Paraboschi S. XML: Current Developments and Future Challenges for the Database Community. In the Proceeding of the 6th International Conference on EDBT, Konstanz, Germany, 2000.3.
- 2 Cheng J, Xu J. IBM DB2 XML Extender — An End-to-end Solution for Storing and Retrieving XML Documents. URL <http://www-3.ibm.com/software/data/db2/extenders/xmlxt/xmlxtbroch.pdf>
- 3 DB2 Developer Domain. Efficient Ways to Publish DB2 Data as XML: Using the New SQL/XML Features in DB2 V8. URL <http://www7b.software.ibm.com/dmdd/>